Realizovane klase:
1. **RSAKeyPairGenerator** – koriscena za generisanje para kljuceva
Metode:
1.1 `public PGPKeyPair generateSignKeyPair(int keySize)` – metoda za generisanje para kljuceva za potpisivanje
1.2 `public PGPKeyPair generateEncryptKeyPair(int keySize)` - metoda za generisanje para kljuceva za enkripciju
1.3 `private KeyPair generateKeyPair(int keySize)` - metoda za generisanje para kljuceva
1.4 `private java.security.KeyPairGenerator initialize(int keySize) throws NoSuchProviderException, NoSuchAlgorithmException` – metoda za inicijalizaciju generatora para kljuceva

2. **Key** – koriscena za model kljuca
2.1 `public Key(Long keyId, String userId)` - konstruktor
2.2 `public Long getKeyId()` - dohvata vrednost keyId
2.3 `public String getUserId()` - dohvata vrednost userId
2.4 `public String toString()` - ispisuje userId, koriscena za testiranja

3. **App** – glavna klasa aplikacije
3.1 `public App()` - konstruktor
3.2 `private void ChooseFileToEncryptClicked(Stage stage, BorderPane root)` - metoda koja se pokrece kada user klikne pokrene odabir fajla za enkriptovanje/potipisivanje
3.3 `private File ChooseFileToDecryptClicked(Stage stage)` – metoda koja se pokrece kada user pokrene odabir fajla za dekripciju/verifikaciju
3.4 `private void ChooseFileForKeyPairImporting(Stage stage, BorderPane pane)` - metoda za odabir fajla prilikom importa kljuc(ev)a
3.5 `private File readFile(Stage stage, FileChooser fileChooser) throws FileNotFoundException` - metoda za citanje odabranog fajla
3.6 `private void GenerateNewKeyPair(BorderPane pane, Stage stage)` – metoda koja pokrece ekran za generisanje novog para kljuceva
3.6 `private String chooseFile(Stage stage)` - genericka metoda za odabir fajla
3.7 `private String formatOutputFileName(File inputFile)` - metoda za formatiranje naziva outpu fajla
3.8 `public void start(final Stage stage)` - start metoda iz Application klase
3.9 `public static void app()` - javna metoda za pokretanje cele aplikacije koja se koristi u Main klasi

4. **Main** – main klasa aplikacije
4.1 `public static void main(String[] args)` – klasicna main metoda java aplikacije

5. **Decryptor** – klasa koja se koristi za dekripciju i/ili verifikaciju fajla
5.1 `public Decryptor(KeyRings keyRings)` - konstruktor
5.2 `private PGPPrivateKey findSecretKey(long keyID)` - metoda za pronalazenje private kljuca za dekripciju
5.3 `public void decryptOrVerifyFile(InputStream in, File outputFile, String signaturePath)` - glavna metoda klase, koristi se kada se pokrene dekripcija ili verifikacija fajla

6. **Encryptor** – klasa skoja se koristi za enkripciju i/ili potpisivanje fajla
6.1 `public Encryptor(KeyRings keyRings)` - konstruktor
6.2 `public void encryptFile(OutputStream out, String filePath, List<Key> recipients, Key signingKey,`

String signPassphrase, boolean integrityCheck, boolean shouldBeCompressed, boolean radix64, EncryptionAlgorithms encryptionAlgorithm) throws IOException, PGPException, NoSuchProviderException, NoSuchAlgorithmException, SignatureException – glavna metoda klase, koristi se za enkripciju ili potpisivanje odabranog fajla

7. **KeyRings** – klasa koja se koristi za rad sa keyring-ovima, generisanje, cuvanje…
7.1 `public KeyRings(KeyPairGenerator keyPairGenerator)` – konstruktor
7.2 `public void generateNewKeyPair(int keySizeSign, int keySizeEncrypt, int encryptionAlgorithm, String userID, String password) throws` PGPException, NoSuchProviderException, NoSuchAlgorithmException – metoda za generisanje para kljuceva
7.3 `public void importKeyPair(InputStream is)` – metoda za import kljuceva
7.4 `public void importPublicKeyRing(PGPPublicKeyRing publicKeyRing)` – metoda koja se koristi ukoliko je potrebno importovati javni kljuc
7.5 `public void importSecretKeyRing(PGPSecretKeyRing secretKeyRing)` – metoda koja se koristi ukoliko je potrebno importovati tajni kljuc
7.6 `public void exportSecretKeyRing(String fileName, String userId)` – metoda koja se koristi za exportovanje tajnog kljuca
7.7 `public boolean exportPublicKeyRing(String fileName, String userID)` – metoda koja se koristi za exportovanje javnog kljuca
7.8 `private static void savePublicKeyRing()` - metoda koja cuva javni keyring u fajl(koristili smo fajl za cuvanje keyring-ova)
7.9 `private static void saveSecretKeyRing()` - metoda koja cuva tajni keyring u fajl
7.10 `public void printPublicKeyRings()` - metoda koja ispisuje ceo javni keyring u konzoli, koriscena za testiranje
7.11 `public void printPrivateKeyRings()` - moteoda koja ispisuje ceo tajni keyring u konzili, koriscena za testiranje
7.12 `private void printPublicKeyRingInfo(PGPPublicKeyRing pkr)` – metoda koja ispisuje jedan javni keyring u konzoli, testiranje
7.13 `private void printSecretKeyRingInfo(PGPSecretKeyRing skr)` - metoda koja ispisuje jedan privatni keyring u konzoli, testiranje
7.14 `public boolean verifySecretKeyPassword(Long keyId, String password)` – metoda koja proverava da li uneta sifra odgovara navedenom kljucu
7.15 `public void deleteSecretKey(Long keyId)` – metoda za brisanje tajnog kljuca
7.16 `public void deletePublicKey(Long keyId)` – metoda za brisanje javnog kljuca
7.17 `public List<Key> getSigningKeys()` - metoda za dohvatanje kljucava za potpisivanje
7.18 `public List<Key> getEncryptionKeys()` - metoda za dohvatanje kljuceva za enkripciju
7.19 `public PGPPublicKeyRingCollection getPublicKeyRings()` - metoda za dohvatanje kolekcije javnih kljuceva
7.20 `public PGPSecretKeyRingCollection getSecretKeyRings()` - metoda za dohvatanje kolekcije tajnih kljuceva
7.21 `public void setKeyPairGenerator(KeyPairGenerator keyPairGenerator)` – metoda za postavljanje zeljenoh generatora kljuceva
7.22 `public PGPPublicKey getEncryptionKey(String userId, Long keyId) throws PGPException` – metoda za dohvatanje jednog kljuca za enkripciju
7.23 `public PGPSecretKey getSigningKey(String userId, Long keyId) throws PGPException` – metoda za dohvatanje jednog kljuca za potpisivanje

8. **Signer** – klasa koriscena za potpisivanje
8.1 `public Signer(KeyRings keyRings)` – konstruktor
8.2 `public void signFile(OutputStream out, String filePath, Key signingKey, String passphrase, boolean radix64) throws PGPException, NoSuchProviderException, NoSuchAlgorithmException, IOException, SignatureException` – glavna metoda klase, koristi se za potpisivanje fajla

9. **Verifier** – klasa koriscena za verifikaciju potpisa
9.1 **public** Verifier(KeyRings keyRings) – konstruktor
9.2 **public void** verifySignature(PGPSignatureList signatureList, String pathToSignature) – glavna metoda klase, koriti se za verifikaciju potpisa fajla

10. **GenerateKey** – frontend klasa, koriscena za postavljane scene za odabir svega sto treba za generisanje kljuca
10.1 **public** GenerateKey(KeyRings keyRings, KeyTable keyTable) – konstruktor
10.2 **public** VBox openAddKeyMenu(BorderPane pane, Stage stage) – metoda koja se koristi spolja za prikaz scene za generisanje para kljuceva
10.3 **private void** createVBox(BorderPane pane, Stage stage) – metoda za postavljanje cele scene I logike

11. **InfoScreen –** klasa koriscena za popup za prikazivanje obavestenja
11.1 **public** InfoScreen(String title, String labelText) – konstruktor

12. **KeyColumn** - klasa za koja predstavlja sve informacije o kljucu koje treba da se prikazu u tabeli
12.1 **public** KeyColumn(String _email, String _name, String _password, **long** _keyId, **boolean** _isPublic, PGPPublicKey _publicKey, PGPSecretKey _secretKey) - konstruktor
12.2 – sva polja imaju sveje getere I setere
12.3 **public long** getOriginalKeyId() - metoda koja dohvata originalan id kljuca, jer se ovde cuva id kljuca u izmenjenom obliku za prikazivanje

13. **KeyTable –** frontend klasa, koriscena za prikazivanje tabele svih kljuceva
13.1 **public** KeyTable(KeyRings keyRings) – konstruktor
13.2 **public** VBox openSecretKeysTable(BorderPane pane, Stage stage) – metoda koja se koristi spolja da bi se otvorila scena za prikazivanje tabele kljuceva
13.3 **private** String getNameFromUserID(String userID) – metoda koja se koristi da se iz userId-a dobije ime user-a(userId ima format “userName <userEmail>”
13.4 **private** String getEmailFromUserID(String userID) – metoda koja se koristi da se iz userId-a dobije email user-a
13.5 **private** List<KeyColumn> getKeyColumns() - metoda koja se koristi za dohvatanje svih kljuceva koji treba da se prikazu(da li je keyPair ili samo javni kljuc)
13.6 **private void** createVBox(BorderPane pane, Stage stage) – metoda koja se koristi da se napravi cela scena I njena logika

14. **PasswordDialog –** klasa koja se koristi za otvaranje dijaloga za unos sifre
14.1 **public** PasswordDialog() - podrazumevani konstruktor
14.2 **public** PasswordDialog(String title, String labelText) – konstruktor

15. **PasswordVerificator** – klasa koja se koristi za proveru sifre, u sebi koristi PasswordDialog kako bi user mogao da unese sifru
15.1 **public static** String verify(**long** keyId, KeyRings keyRings) – glavna I jedina metoda klase, koristi se otvaranje PasswordDialog-a I proveru ispravnosti unete sifre

16. **SignAndEncrupt –** frontend klasa, koriscena za prikaz scene za enkripciju I potpisivanje
16.1 **public** SignAndEncrypt(KeyRings keyRings) – konstruktor
16.2 **public** VBox openSignAndEncrypt(BorderPane pane, Stage stage, KeyTable keyTable, String filePath, KeyRings keyRings) - metoda koja se koristi spolja za otvaranje scene za enrkipciju I potpisivanje
16.3 **private void** createVBox(BorderPane pane) – metoda koja se koristi za kreiranje scene za enkripciju I potpisivanje